

Atty. Docket No. 03AB014C

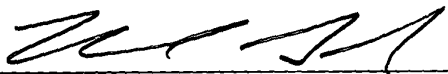
SYSTEM AND METHODOLOGY PROVIDING
AUTOMATION SECURITY ANALYSIS,
VALIDATION, AND LEARNING IN AN INDUSTRIAL
CONTROLLER ENVIRONMENT

by

David D. Brandt, Kenwood Hall,
Mark Burton Anderson, Craig D. Anderson,
and George Bradford Collins

MAIL CERTIFICATION

I hereby certify that the attached patent application (along with any other paper referred to as being attached or enclosed) is being deposited with the United States Postal Service on this date September 12, 2003, in an envelope as "Express Mail Post Office to Addressee" Mailing Label Number EV330022816US addressed to the Mail Stop Patent Application, Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.



Himanshu S. Amin

TITLE: SYSTEM AND METHODOLOGY PROVIDING AUTOMATION SECURITY
ANALYSIS, VALIDATION, AND LEARNING IN AN INDUSTRIAL
5 CONTROLLER ENVIRONMENT

REFERENCE TO RELATED APPLICATION(S)

This application claims the benefit of U.S. Provisional Patent Application Serial
No. 60/420,006 which was filed October 21, 2002, entitled System and Methodology
10 Providing Automation Security in an Industrial Controller Environment, the entirety of
which is incorporated herein by reference.

TECHNICAL FIELD

The present invention relates generally to industrial control systems, and more
15 particularly to a system and methodology to facilitate electronic and network security in
an industrial automation system.

BACKGROUND OF THE INVENTION

Industrial controllers are special-purpose computers utilized for controlling
20 industrial processes, manufacturing equipment, and other factory automation, such as data
collection or networked systems. In accordance with a control program, the industrial
controller, having an associated processor (or processors), measures one or more process
variables or inputs reflecting the status of a controlled system, and changes outputs
effecting control of such system. The inputs and outputs may be binary, (e.g., on or off),
25 as well as analog inputs and outputs assuming a continuous range of values.

Measured inputs received from such systems and the outputs transmitted by the
systems generally pass through one or more input/output (I/O) modules. These I/O
modules serve as an electrical interface to the controller and may be located proximate or
remote from the controller including remote network interfaces to associated systems.

Inputs and outputs may be recorded in an I/O table in processor memory, wherein input values may be asynchronously read from one or more input modules and output values written to the I/O table for subsequent communication to the control system by specialized communications circuitry (*e.g.*, back plane interface, communications module). Output modules may interface directly with one or more control elements, by receiving an output from the I/O table to control a device such as a motor, valve, solenoid, amplifier, and the like.

At the core of the industrial control system, is a logic processor such as a Programmable Logic Controller (PLC) or PC-based controller. Programmable Logic Controllers for instance, are programmed by systems designers to operate manufacturing processes *via* user-designed logic programs or user programs. The user programs are stored in memory and generally executed by the PLC in a sequential manner although instruction jumping, looping and interrupt routines, for example, are also common. Associated with the user program are a plurality of memory elements or variables that provide dynamics to PLC operations and programs. These variables can be user-defined and can be defined as bits, bytes, words, integers, floating point numbers, timers, counters and/or other data types to name but a few examples.

Various remote applications or systems often attempt to update and/or acquire PLC information or related device information *via* a plurality of different, competing and often incompatible or insecure network technologies. A major concern with this type of access to PLC's and control systems in general, relates to the amount of security that is provided when sending or receiving data to and from the PLC and/or associated equipment. In most factories or industrial environments, complex and sometimes dangerous operations are performed in a given manufacturing setting. Thus, if a network-connected controller were inadvertently accessed, or even worse, intentional sabotage were to occur by a rogue machine or individual, potentially harmful results can occur.

One attempt at providing security in industrial control systems relates to simple password protection to limit access to the systems. This can take the form of a plant or controls Engineer or Administrator entering an alpha-numeric string that is typed by an

operator each time access is attempted, wherein the controller grants access based on a successful typing of the password. These type passwords are highly prone to attack or discovery, however. Often times, users employ passwords that are relatively easy to determine (*e.g.*, person's name or birthday). Sometimes, users exchange passwords with other users, whereby the password is overheard or simply, a user with improper authorization comes in contact with the password. Even if a somewhat higher level of security is provided, parties employing sophisticated hacking techniques can often penetrate sensitive control systems, whereby access should be limited to authorized users and/or systems in order to mitigate potentially harmful consequences.

SUMMARY OF THE INVENTION

The following presents a simplified summary of the invention in order to provide a basic understanding of some aspects of the invention. This summary is not an extensive overview of the invention. It is intended to neither identify key or critical elements of the invention nor delineate the scope of the invention. Its sole purpose is to present some concepts of the invention in a simplified form as a prelude to the more detailed description that is presented later.

The present invention relates to a system and methodology to facilitate network and/or automation device security in an industrial automation environment. Various systems and methodologies are provided to promote security across and/or within networks and in accordance with different automation device capabilities. In one aspect of the present invention, a Security Analysis Methodology (SAM) and tool provides an automated process, component, and tool that generates a set (or subset) of security guidelines, security data, and/or security components. An input to the tool can be in the form of an abstract description or model of a factory, wherein the factory description includes one or more assets to be protected and associated pathways to access the assets. Security data generated by the tool includes a set of recommended security components, related interconnection topology, connection configurations, application procedures, security policies, rules, user procedures, and/or user practices, for example.

SAM can be modeled on a risk-based/cost-based approach, if desired. A suitable level of protection can be determined to facilitate integrity, privacy, and/or availability of assets based on risk and/or cost. In addition, descriptions of shop floor access, Intranet access, Internet access, and/or wireless access can also be processed by the tool. Since
5 multiparty involvement can be accommodated (IT, Manufacturing, Engineering, *etc.*), the tool can be adapted for partitioned security specification entry and sign-off. The security data of the SAM tool can be generated in a structured security data format (*e.g.*, XML, SQL) that facilitates further validation and compliance checking of the security data, if desired.

10 In another aspect of the present invention, a security Validation Methodology and associated tools can be provided. The validation tools perform initial and periodic live security assessment of a physical system. This enables security flaws or weaknesses to be identified. One aspect of the tools is to check a system prior to security modifications in order to assess current security levels. Another aspect is to check a system for
15 conformance – either to recommendations of a security analysis, and/or against standards such as ISO, for example. The validation tools can be executed on end devices (host based), and/or executed as an independent device that is operatively coupled to a network (network based) at selected points. One function of host-validation tools is to perform vulnerability scanning and/or auditing on devices. This includes revision checks,
20 improper configuration check, file system/registry/database permissions check, user privilege/password and/or account policy checks, for example.

One function of the network validation tools is to perform vulnerability scanning and auditing on the networks. This includes checking for susceptibility to common network-based attacks, searching for open TCP/UDP ports, and scanning for vulnerable
25 network services. The tools can also attempt to gain key identity information about end devices that may enable hacker entry. Another function of the network validation tools is to perform vulnerability scanning and auditing on firewalls, routers, and/or other network/security devices. In addition, a complementary tool can be provided to assess CIP-based factory automation systems for security. This will typically be a network-based

tool, since factory automation devices often are not as capable as general purpose computing devices. The tool can also be operable in an assessment mode to discover system flaws with little or no configuration, and the tool can operate in a validation mode to check system security against security analysis methodology determinations described above. Still yet other functions can include non-destructively mapping a topology of IT and automation devices, checking revisions and configurations, checking user attributes, and/or checking access control lists. The validation tools described herein can also be adapted to automatically correct security problems (*e.g.*, automatically adjust security parameters/rules/policies, install new security components, remove suspicious components, and so forth).

According to another aspect of the present invention, a Security Learning system is provided that can include network-based aspects and/or host-based aspects and similar to some of the security aspects described above with respect to the Validation tools. A network-based security learning system (also referred to as learning component) is provided that monitors an automation network during a predetermined training period (*e.g.*, monitor network activities for 1 week). During the training period, the learning component monitors and learns activities or patterns such as: the number of network requests to and from one or more assets; the type of requests (*e.g.*, read/write, role/identity of person/system requesting access, time of requests); status or counter data (*e.g.*, network access counters, error codes) which can be provided or queried from a learning or status component within the asset; and/or monitor and learn about substantially any data type or pattern that may be retrieved from the network and/or the asset.

After the training period, the learning component monitors the automation network and/or assets for detected deviations from data patterns learned during the training period. If desired, a user interface can be provided, wherein one or more pattern thresholds can be adjusted (also can provide options for the type of data patterns to monitor/learn). For example, if the number of network requests to the asset has been monitored and learned to be about 1000 requests per hour during the past month, then a

threshold can be set *via* the user interface that triggers an alarm or causes an automated event to occur if a deviation is detected outside of the threshold (*e.g.*, automatically disable all network requests from the other networks if the number of network requests to the asset exceeds 10% of the average daily network requests detected during the training period).

Various learning functions and/or processes can be provided to facilitate automated learning within the learning components. This can include mathematical processes, statistical processes, functions, and/or algorithms and include more elaborate systems such as a neural network, for example. In addition, artificial intelligence functions, components and/or processes can be provided. Such components can include automated classifiers for monitoring and learning data patterns, wherein such classifiers include inference models, Hidden Markov Models (HMM), Bayesian models, Support Vector Machines (SVM), vector-based models, decision trees, and the like.

The following description and the annexed drawings set forth in detail certain illustrative aspects of the invention. These aspects are indicative, however, of but a few of the various ways in which the principles of the invention may be employed and the present invention is intended to include all such aspects and their equivalents. Other advantages and novel features of the invention will become apparent from the following detailed description of the invention when considered in conjunction with the drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a schematic block diagram illustrating automation security tools in accordance with an aspect of the present invention.

Fig. 2 is a schematic block diagram illustrating a security analysis tool in accordance with an aspect of the present invention.

Fig. 3 is a diagram illustrating an example security analyzer in accordance with an aspect of the present invention.

Fig. 4 is a diagram illustrating an example security analysis schema in accordance with an aspect of the present invention.

Fig. 5 is a diagram illustrating a validation system, methodology, and security validation tools in accordance with an aspect of the present invention.

Fig. 6 is a schematic block diagram illustrating a validation analyzer in accordance with an aspect of the present invention.

5 Fig. 7 is a schematic block diagram illustrating a security learning system in accordance with an aspect of the present invention.

Fig. 8 is a diagram illustrating a learning component in accordance with an aspect of the present invention.

10 Fig. 9 is a schematic block diagram illustrating a learning analyzer in accordance with an aspect of the present invention.

Fig. 10 is a flow diagram illustrating security analysis processing in accordance with an aspect of the present invention.

Fig. 11 is a flow diagram illustrating security validation processing in accordance with an aspect of the present invention.

15 Fig. 12 is a flow diagram illustrating security learning and detection processing in accordance with an aspect of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

20 The present invention relates to a system and methodology facilitating automation security in a networked-based industrial controller environment. Various components, systems and methodologies are provided to facilitate varying levels of automation security in accordance with security analysis tools, security validation tools and/or security learning systems. The security analysis tool receives abstract factory models or descriptions for input and generates an output that can include security guidelines,
25 components, topologies, procedures, rules, policies, and the like for deployment in an automation security network. The validation tools are operative in the automation security network, wherein the tools perform security checking and/or auditing functions, for example, to determine if security components are in place and/or in suitable working order. The security learning system monitors/learns network traffic patterns during a

learning phase, fires alarms or events based upon detected deviations from the learned patterns, and/or causes other automated actions to occur.

It is noted that as used in this application, terms such as “component,” “tool,” “analyzer,” and the like are intended to refer to a computer-related entity, either
5 hardware, a combination of hardware and software, software, or software in execution as applied to an automation system for industrial control. For example, a component may be, but is not limited to being, a process running on a processor, a processor, an object, an executable, a thread of execution, a program and a computer. By way of illustration, both an application running on a server and the server can be components. One or more
10 components may reside within a process and/or thread of execution and a component may be localized on one computer and/or distributed between two or more computers, industrial controllers, and/or modules communicating therewith.

Referring initially to Fig. 1, a system 100 illustrates various automation security tools in accordance with an aspect of the present invention. One or more automation
15 assets 120 communicate and cooperate with various network devices 124 across a network 130. The automation assets 120 include substantially any type of control, communications module, computer, I/O device, Human Machine Interface (HMI) that communicate *via* the network 130 which includes control, automation, and/or public networks. In one example, the automation assets 120 include Programmable Logic
20 Controllers (PLC) that can also communicate to and control various other assets such as Input/Output modules including Analog, Digital, Programmed/Intelligent I/O modules, other programmable controllers, communications modules, and the like. The network 130 includes public networks such as the Internet, Intranets, and automation networks such as Control and Information Protocol (CIP) networks including DeviceNet and
25 ControlNet. Other networks 130 include Ethernet, DH/DH+, Remote I/O, Fieldbus, Modbus, Profibus, wireless networks, serial protocols, and so forth. In addition to the automation assets 120, the network devices 124 include various possibilities (hardware and/or software components). These include components such as switches with virtual local area network (VLAN) capability, LANs, WANs, proxies, gateways, routers,

firewalls, virtual private network (VPN) devices, intrusion detection systems, servers, clients, computers, configuration tools, monitoring tools, and/or other devices.

According to one aspect of the present invention, various security tools can be provided with the system 100. Although three tools are illustrated, it is to be appreciated that more or less than three tools can be employed with the present invention and in a plurality of similar or different combinations. In one aspect, a security analysis tool 140 is provided that receives factory input data 144 describing or modeling various aspects of the automation assets 120, network devices 124, network 130, and/or system 100. The security analysis tool 140 processes the factory input data 144 and generates security output data 150 which is then deployed to machines and/or users in order to facilitate suitable network security measures and practices in the system 100. As will be described in more detail below, such measures can include security recommendations, configuration guidelines or adjustments, procedures, rules, policies, and security parameters, for example, that are utilized to mitigate unwanted intrusions or attacks from the network 130 that may affect the automation assets 120 and/or network devices 124.

In another aspect of the present invention, one or more validation tools 160 can be provided (can be host and/or networked based) that perform automated security auditing and checking functions on the network 130, the automation assets 120, and/or network devices 124 to determine if suitable security standards have been implemented. The validation tools also perform periodic or monitored assessments within the system 100 to determine if potential network threats or attacks are at hand. As will be described in more detail below, this can include automated and/or healing operations to mitigate network security threats. In another aspect of the present invention, one or more learning tools 170 can be provided (can also be host and/or networked based) that learn system activities or patterns during a training or configuration period, then perform automated actions in response to detected deviations from the learned activities or patterns. Such automated actions can include altering network activity (*e.g.*, preventing further network attempts to automation assets or network devices) and firing an alarm such as an e-mail or pager to notify an entity (user and/or machine) of a potential or detected problem.

It is noted that the security tools 140, 150 and/or 160 can share or exchange information between tools. For example, the security analysis tool 140 can receive input from the validation tool 160 (*e.g.*, three new network devices detected in topology), wherein the security analysis tool generates new or adjusted security output data 150 in response thereto. It is further noted that one or more of the automation assets 120 may directly access the network 130 and/or may employ the network devices 124 to achieve network access.

Turning to Fig. 2, a security analysis tool 200 is illustrated in accordance with an aspect of the present invention. The security analysis tool 200 operates on a computer or workstation and receives one or more factory inputs 210 that can be generated from a plurality of sources. Such sources can include user input, model input (*e.g.*, asset models, network models), schemas, formulas, equations, maps, and codes, for example. The factory inputs 210 are then processed by the security analysis tool 200 to generate one or more security outputs 220 which can also be provided in various forms such as manuals, documents, schemas, executables, codes, e-mails, and/or other electronic data that is described in more detail below. As illustrated, a Graphical User Interface 230 (GUI) or interface application can be provided to interact with the security analysis tool 200, factory inputs 210, and/or security outputs 220. This can include substantially any type of application that sends, retrieves, processes, and/or manipulates factory input data 210, receives, displays, formats, and/or communicates security output data 220, and/or facilitates operation of the security analysis tool 200. For example, such interfaces 230 can also be associated with an engine, editor tool or web browser although other type applications can be utilized. The GUI 230 includes a display 234 having one or more display objects (not shown) including such aspects as configurable icons, buttons, sliders, input boxes, selection options, menus, tabs and so forth having multiple configurable dimensions, shapes, colors, text, data and sounds to facilitate operations with the security analysis tool 200. In addition, the GUI 230 can also include a plurality of other inputs 240 or controls for adjusting and configuring one or more aspects of the present invention. This can include receiving user commands from a mouse, keyboard, speech

input, web site, remote web service and/or other device such as a camera or video input to affect or modify operations of the GUI 230.

Referring now to Fig. 3, an example security analyzer 300 is illustrated in accordance with an aspect of the present invention. The security analyzer 300 can be an automated process, application, component, and/or tool that generates a set of security guidelines or security data 310 and executes a Security Analysis Method (SAM) in accordance with the present invention. An input to the security analyzer 300 is an abstract description of a factory depicted as factory data 320. The factory data 320 can describe or model one or more automation assets to be protected and associated network pathways to access the assets. Other factory data 320 can include risk data, cost data, security feedback from other security tools, network access patterns, and partitioning data, for example. Security data 310 generated by the security analyzer 300 includes a set of recommended security components, related interconnection topology, connection configurations, application procedures, security policies, rules, user procedures, and/or user practices, for example, that is employed to guide users and adapt systems with various security measures.

The Security Analysis Method noted above, and security analyzer 300 can also be modeled on a risk-based/cost-based approach, if desired. A suitable level of protection can be determined to facilitate integrity, privacy, and/or availability of assets based on risk and/or cost. Thus, security parameters, policies, and procedures, for example, can be increased if lower security risks and associated costs are desired, whereas security measures can be decreased if higher risks and/or costs associated with network attacks or intrusions are deemed acceptable. In addition, descriptions of shop floor access, Intranet access, Internet access, wireless access and/or other network access patterns can also be described as factory data 320 and processed by the security analyzer 300. Since multiple party involvement can be accommodated (*e.g.*, IT, Manufacturing, Engineering, *etc.*), the security analyzer 300 can be adapted for partitioned security specification entry and sign-off. The security data 310 can be generated in a structured security data format (*e.g.*, XML, SQL) that facilitates further validation and compliance checking of the security

data, if desired. As illustrated, a security analysis schema 330 which is described in more detail below, can be derived from the security data 310 and can be provided to other entities such as users or machines for further security processing/implementations.

Fig. 4 illustrates an exemplary schema that may be employed for security
5 deployments, communications, and configurations in accordance with the present invention. Although the schema represents one possible manner in which to transfer data to and from an entity such as a user, interface, file, an automation component and associated network devices, it is to be appreciated that other possible data transfer mechanisms may be employed. For example, data can be transmitted in the form of
10 binary or other type data packets that convey information in accordance with the present invention.

Referring to Fig. 4, an example security analysis schema 400 is illustrated in accordance with an aspect of the present invention. The security analysis schema 400 includes one or more XML elements 410 through 440 (defined by starting and ending
15 tags with (</> symbols), arranged in substantially any order) that relate to one or more security items or data and provide information to facilitate security guidelines and configurations. Although not shown, the XML elements and associated tags can also include attribute information if desired, wherein an attribute is a name-value pair associated with an element start tag (e.g., <topology = "PLC connected to gateway device
20 having firewall protection">). The security analysis schema 400 can then be deployed to various systems and/or components to control/adapt network access based upon the security contents specified therein.

Proceeding to 410, a recommendations element can be provided having associated recommendations data. This can include suggestions as to how to adapt automation
25 components and network devices for suitable security measures (e.g., in view of risk and cost criteria). In one example, a suggestion can be in the form of a statement "All real time control devices and networks should only be connected to public networks *via* front-end server having virus detection, intrusion detection, and virtual private network capabilities." In another example, "Remote factory network devices must be identified,

authorized, and authenticated before achieving access to control network, otherwise, local factory network devices should communicate with low-end encryption technologies.” As can be appreciated, a plurality of such recommendations can be provided. At 414, a topologies element can be provided. This can include information on how to interconnect various devices and networks to achieve desired security goals (*e.g.*, PLC connects to router, router connects to factory server and protected gateway...). In another aspect, the topology data 414 can be in the form of symbols or codes that are employed to construct topology or network maps/displays *via* a visual or other type application.

At 420, configuration data can be provided. This type of data can include settings or parameters for adapting network components with suitable security measures (*e.g.*, communications module word three should be set to value 03AA Hex for extended security checking, set dip switch two on gateway to cause authentication and authorization procedures with outside network devices, install virus detection software on network server...). In another aspect, the configuration data can be sent or deployed to devices *via* the schema 400 and loaded to cause automatic configurations. At 424, an applications procedure element can be provided having associated procedure data. Such data can include the types of security applications to load, any security adjustments or settings relating to the applications, application status information to verify, and procedures for correctly operating respective security applications to mitigate potential attacks or threats.

At 430, policy data can be provided. The policy can be general and/or specific, applied system wide and/or to a device or subset of devices. For example location-based policies can be initiated (*e.g.*, all network requests from listed URL’s are to be denied, network requests from Pittsburgh server limited to 100 per day). Time-based policies can also be defined (*e.g.*, no outside network requests allowed between 10:00AM and 2:00PM). Process-based policies can be defined such as for example, “Limit outside network requests to below 50 during real time batch operations.” Other policies include load-based policies, whereby network requests that are responded to are regulated in accordance with an amount of desired network traffic (*e.g.*, regulated according to

requests/hour). Other policies may be related to the type of requests (*e.g.*, all requests to write data to the PLC are to be denied, outside devices cannot update analog module configuration data, communications module to provide status data only). In general, substantially any policy that defines, regulates, and/or limits network activities in view of security considerations can be employed with the present invention.

At 434, one or more security rules can be provided that have similar effects as the policies described above. For example, rules can be provided in an If / Then construct (can include else, else if, Boolean expressions and the like), wherein if a defined condition or conditions occur, then one or more listed actions result (can include nested constructs) (*e.g.*, If more than 3 network access attempts are negotiated unsuccessfully, then deny further communications with node or address). At 440, user procedure data can be provided. This can include actual procedure data and/or links to databases or websites to acquire the data. Such data can instruct users on suitable security procedures, security precautions, training, configurations, examples, wizards, manuals, trouble shooting, emergency contacts, contact information, maintenance, and the like which are designed to mitigate system security problems.

Fig. 5 illustrates a validation system 500, methodology, and validation tools 550, 560 in accordance with an aspect of the present invention. The validation tools 550 and 560 perform initial and periodic live security assessment of a physical system. This enables security flaws or weaknesses to be identified. One aspect of the tools is to check the system 500 prior to proposed or attempted security modifications in order to assess current security levels. Another aspect is to check the system 500 for conformance – to the recommendations of a security analysis tool described above, and/or against standards such as ISO, for example.

The validation tools 550 and 560 can be executed on end devices 570 (host based), and/or executed as an independent device 580 that is attached to a network 590 (network based) at selected points. One function of the host-validation tool 550 is to perform vulnerability scanning and/or auditing on devices. This includes revision checks, improper configuration check, file system/registry/database permissions check, user

privilege/password and/or account policy checks, for example.

One function of the network validation tool 560 is to perform vulnerability scanning and auditing on the networks 590. This includes checking for susceptibility to common network-based attacks, searching for open TCP/UDP ports, and scanning for
5 vulnerable network services. The tools 550 and 560 can also attempt to gain key identity information about end devices that may enable hacker entry.

Another function of the network validation tool 560 is to perform vulnerability scanning and auditing on firewalls, routers, and/or other security devices. In addition, a complementary tool can be provided to assess CIP-based factory automation systems for
10 security (includes substantially any factory protocol). This will typically be a network-based tool, since factory automation devices often are not as capable as general purpose computing devices. The network validation tool 560 can also be operable in an assessment mode to discover system flaws with little or no configuration, and the tool can operate in a validation mode to check system security against security analysis
15 methodology determinations described above. Still yet other functions can include non-destructively mapping a topology of IT and automation devices, checking revisions and configurations, checking user attributes, and/or checking access control lists. The validation tools described herein can also be adapted to automatically correct security problems (*e.g.*, automatically adjust security parameters, install new security components,
20 remove suspicious components, and so forth). It is to be appreciated that one or more of the functions described herein for the host validation tool 550 may be shared/interchanged with the network validation tool 560, and *visa versa*.

Referring now to Fig. 6, a validation analyzer 600 is illustrated in accordance with an aspect of the present invention. The validation analyzer 600 can be a hardware device,
25 computer, processor, application, and/or combination thereof that process one or more security data inputs 610 such as can be received or communicated from a network (not shown). The security data inputs 610 include current security data, network data, audit data, device data, security analysis data, and/or other data that can be derived from scanning or querying a network and associated devices *via* the validation analyzer 600 for

information regarding current network security conditions. Various components can be provided with the validation analyzer 600 to facilitate security monitoring and processing. In one aspect, an assessment component 620 can be provided. The assessment component 620 performs initial and/or periodic security determinations on network systems to identify security deficiencies or problems therein. For example, the assessment component 620 may compare a stored security configuration with a network configuration received from the security data inputs 610, flag such conditions, and/or institute further actions if differences are detected.

In another aspect, a standards component 624 can be provided to perform security compliance checking. This can include automated checking prior to proposed or attempted network security modifications in order to assess current security levels. Compliance checking can also include determining conformance to other automated security analysis recommendations, conformance to applicable device/network security standards, and/or in accordance with predetermined or factory-specific standards, for example. Such checking can be in accordance with stored standards or procedures within the validation analyzer 600, or can include remote checking to such resources as network databases, web sites, web services (*e.g.*, databases linked to Internet Protocol Security Standard, IEEE database). It is noted that the assessment component 620 and/or standards component 624 can initiate vulnerability scanning and/or auditing on devices/networks/systems. This can include revision checks, improper configuration checks, file system/ registry/ database permissions checks, user privilege/password and/or account policy checks, checking for susceptibility to network-based attacks, searching for open network ports, scanning for vulnerable network services, learning identity information about end devices/users that may enable attack entry, performing vulnerability scanning and auditing on firewalls, routers, and/or other security devices or components, non-destructively mapping a topology of network devices, checking revisions and configurations, checking user attributes, and/or checking network/device access control lists. As can be appreciated, such checking can include comparisons to local/remote databases or sites as noted above.

In yet another aspect of the present invention, a learning/analyzer component 628 can optionally be provided within the validation analyzer 600. This component can be adapted to learn network, device, and/or system patterns, scan current network data, and process the current network data in accordance with the learned patterns to possibly
5 initiate other automated actions. The learning/analyzer component 628 will be described in more detail below with respect to Figs. 7-9.

If a security issue or problem is detected by the assessment component 620, standards component 624, and/or learning/analyzer component 628, a flag or event can be fired that triggers an automated action component 650, wherein one or more automated
10 security actions can be initiated. The automated security actions can include automatically correcting security problems at 654 such as automatically adjusting security parameters, altering network traffic patterns at 658 (*e.g.*, increasing/decreasing communications with a node), installing new security components and/or removing/disabling suspicious components at 662, firing alarms, and/or automatically
15 notifying entities about detected problems and/or concerns at 670, and/or generating security data at 674 such as generating an error or log file, generating a schema, generating data to re-configure or re-route network connections, updating a database or remote site, for example. As illustrated, the validation analyzer 600 can be configured and interacted with *via* a user interface 680 having similar input and output functionality
20 as described above with respect to the user interface depicted in Fig. 2.

Fig. 7 illustrates a security learning system 700 in accordance with an aspect of the present invention. The security learning system 700 that can include network-based aspects and/or host-based aspects and similar to some of the security aspects described above with respect to Fig. 5. A network-based security learning system 710 (also referred
25 to as learning component 710) is provided that monitors an automation network 714 during a predetermined training period (*e.g.*, monitor network activities for 1 month).

During the training period, the learning component 710 monitors and learns activities or patterns such as:

- The number of network requests to and from one or more assets 720;

- the type of requests (*e.g.*, read/write, role/identity of person/system requesting access, time of requests);
 - status or counter data (*e.g.*, network access counters, error codes) which can be provided or queried from a learning or status component 724 within the asset 720;
- 5 and/or
- monitor and learn about substantially any data type or pattern that may be retrieved from the network 714 and/or the asset 720.

Network activities can also include network requests that are received from outside networks 730 that may be routed through a security gateway or server 734 before reaching the automation network 714.

After the training period, the learning component 710 monitors the automation network 714 and/or assets 720 for detected deviations from data patterns learned during the training period. If desired, a user interface (not shown) can be provided, wherein one or more pattern thresholds can be adjusted (also can provide options for the type of data patterns to monitor/learn). For example, if the number of network requests to the asset 720 has been monitored and learned to be about 1000 requests per hour during the past month, then a threshold can be set *via* the user interface that triggers an alarm or causes an automated event to occur if a deviation is detected outside of the threshold (*e.g.*, automatically disable all network requests from the other networks 730 if the number of network requests to the asset 720 exceeds a set or determined percentage of the average daily network requests detected during the training period).

In one aspect, the learning component 710 and associated detection parameters or thresholds can be provided as a network-based tool or tools that can reside at various portions of the automation network 714. In another aspect, the learning component can be provided as a host-based component as illustrated at 724 – depending on the resources available for the asset 720.

Various learning functions and/or processes can be provided to facilitate automated learning within the learning components 710 and 724. This can include mathematical processes, statistical processes, functions, and/or algorithms and include

more elaborate systems such as a neural network, for example. In addition, artificial intelligence functions, components and/or processes can be provided. Such components can include automated classifiers for monitoring and learning data patterns, wherein such classifiers include inference models, Hidden Markov Models (HMM), Bayesian models, Support Vector Machines (SVM), vector-based models, decision trees, and the like.

Fig. 8 illustrates a learning component 800 in accordance with an aspect of the present invention. The learning component 800 can be configured with various data types, circuits, algorithms, applications, and so forth that are adapted to learn from data or events generated from a training data set 810. The training data set 810 is derived by monitoring network or device activities over a predetermined timeframe. Such activities include network events, network data, network device activities, automations asset activities, and monitoring status information, for example. The activities can also include network access patterns, network attempts, network sources, data transfer and exchange activities, network/device load considerations, time considerations, and location considerations, for example (*e.g.*, what time does heaviest network traffic occur, where do most network requests originate, what regions do most hacking attempts originate).

In order to process the training data 810, the learning component 800 includes one or more learning models 820 and/or learning variables 830. As noted above, the learning models 820 can include such aspects as neural network functions, inference models, mathematical models, statistical models, probabilistic models, classifiers, and so forth that learn network patterns or occurrences from the training data 810. It is also noted that the learning models can be adapted similarly (*e.g.*, all models configured as Hidden Markov Models) or adapted in various combinations (*e.g.*, 40 models configured as a neural network, 3 models adapted in a Bayesian configuration, 1 model configured as a vector-based classifier). The learning variables 830 can be focused on selected events or circumstances. For example, a network load variable may record the average number of outside network requests per hour. In another example, a PLC variable may record the average number of network retries that an associated PLC experiences in a given timeframe, whereas another PLC variable records the maximum number of network

retrieves that the PLC experienced during the same timeframe. In another aspect, the learning variables 820 may be employed as counters to record amounts for various events (e.g., record the number of PLC network transfers to I/O device over the last hour). As can be appreciated, a plurality of such variables can be defined and updated to log various network events during a selected training period. After training, the learning component 810 stores learned patterns or events that are then employed by a learning analyzer component described below to monitor and detect network security problems or identify potential security issues.

Fig. 9 illustrates a learning analyzer 900 in accordance with an aspect of the present invention. The learning analyzer 900 monitors current network and/or device data 910, determines whether the current data is within tolerance of historical data patterns that were previously learned/recorded, and initiates one or more automated actions 920 if current data 910 including trends derived therefrom are determined outside of the tolerance. These determinations can be achieved *via* a comparison analyzer 930 that compares learned data patterns with current data patterns 944 in accordance with threshold and/or range data illustrated at 950. For example, a learned pattern 940 could be that between 11:00 and 12:00, network load between four network devices is about ten million data packet transfers. Thus, if a threshold 950 were set for one million transfers, and if current data patterns 944 exceeded more than one million transfers above the learned data patterns 940 (ten million transfers during the selected period), then the comparison analyzer 930 would detect this overload (e.g., *via* subtraction of current and learned data, then comparing to threshold data) and initiate the automated actions 920.

Similar to the validation components described above, the automated actions 920 can include automatically correcting security problems such as automatically adjusting security parameters, altering network traffic patterns, installing new security components, removing/disabling suspicious components, firing alarms, and/or automatically notifying entities about detected problems and/or concerns among other actions, for example.

In another aspect, the threshold data 950 can include range data thus providing upper and lower thresholds for given patterns. For example, a range can be specified to

detect events that occur within or outside the selected range. In the example above, a range may have been specified as plus and minus one million transfers (do not have to be equidistant ranges), thus if current data patterns were detected to be above eleven million or below nine million transfers, then an automated action 920 would be initiated by the comparison analyzer 930 if current data patterns were outside the selected range of 10 million, +/- 1million transfers. As can be appreciated, a plurality of thresholds and/or ranges 950 can be specified. In addition, the threshold and range data 950 can be specified in various formats (*e.g.*, in accordance with standard deviation), and can include dynamically adjustable thresholds or ranges (*e.g.*, set threshold high in the morning and lower in the afternoon, change threshold according to real time processing requirements).

As illustrated, the comparison analyzer 930 can also monitor, analyze, and detect deviations of stored variables 960 and current variables 964 in view of the threshold and range data 950. A user interface 970, having similar display/input functionality as previously described, can be provided to specify/adjust the threshold and/or range data 950. The user interface 970 can also interact with and control the learning analyzer 900 (*e.g.*, set threshold or ranges, add, remove, adjust learning models, view analyzer status, configure automated actions, monitor variables, adjust variables, generate security reports and the like).

Figs. 10-12 illustrate security methodologies in accordance with an aspect the present invention. While, for purposes of simplicity of explanation, the methodologies are shown and described as a series of acts, it is to be understood and appreciated that the present invention is not limited by the order of acts, as some acts may, in accordance with the present invention, occur in different orders and/or concurrently with other acts from that shown and described herein. For example, those skilled in the art will understand and appreciate that a methodology could alternatively be represented as a series of interrelated states or events, such as in a state diagram. Moreover, not all illustrated acts may be required to implement a methodology in accordance with the present invention.

Fig. 10 illustrates a security analysis process 1000 in accordance with an aspect of the present invention. Proceeding to 1010, factory descriptions of automation assets, network devices, network topologies, and/or other factory data are generated. Such data can include an abstract description of a factory, models, equations, maps, and network pathways to access the automation assets. The descriptions can also include risk data, cost data, security data from other security tools, and partitioning or user data, for example. At 1018, the factory descriptions are processed such as *via* an object, application, security engine, ASIC, computer, web service, and so forth.

At 1022, security output data is determined in accordance with the factory descriptions and processing described above. The security output data can include a set or subset of recommended security components, codes, parameters, settings, related inter-connection topology, connection configurations, application procedures, security policies, rules, user procedures, and/or user practices, for example, as noted above. At 1026, security output data is generated that can be automatically deployed to one or more entities such as users or devices in order to implement various security measures within an automation environment (*e.g.*, data file or schema generated to automatically configure devices, provide user training and precautions, provide security configurations and topologies). At 1030, when the security output data has been disseminated, entities employ the security data to mitigate network security issues such as unwanted network access and/or network attack.

Fig. 11 illustrates a security validation process 1100 in accordance with an aspect of the present invention and includes host-based and/or network based processing as noted above. Proceeding to 1110, security assessments are performed. This can include initial and/or periodic live security assessment of a physical system to identify security flaws or weaknesses. At 1122, security compliance tests are performed. This can include automated checking prior to proposed or attempted network security modifications in order to assess current security levels. Compliance checking can also include determining conformance to other automated security analysis recommendations,

conformance to applicable device/network security standards, and/or in accordance with predetermined or factory-specific guidelines, for example.

At 1126, vulnerability scanning and/or auditing on devices/networks is performed. This includes revision checks, improper configuration checks, file system/ registry/
5 database permissions checks, user privilege/password and/or account policy checks, checking for susceptibility to common network-based attacks, searching for open network ports, scanning for vulnerable network services, learning identity information about end devices/users that may enable hacker entry, performing vulnerability scanning and auditing on firewalls, routers, and/or other security devices, non-destructively mapping a topology
10 of IT and automation devices, checking revisions and configurations, checking user attributes, and/or checking access control lists. At 1124, a determination is made as to whether security issues have been detected such as in accordance with the assessments, compliance testing, and scanning/auditing described above. If no security issues are detected at 1124, the process proceeds back to 1110. If security issues are detected at
15 1130, the process proceeds to 1134. At 1134, one or more automated security actions are performed to mitigate security threats. This can include automatically correcting security problems such as automatically adjusting security parameters, altering network traffic patterns, installing new security components, removing suspicious components, firing alarms, and/or automatically notifying entities about detected problems and/or suspicions.
20 After automated processing at 1134, the process proceeds back to 1110 for further security processing, analysis, scanning, and detection.

Fig. 12 illustrates a security learning and detection process 1200 in accordance with an aspect of the present invention and can also include network-based aspects and/or host-based aspects as noted above. At 1210, one or more learning components such as
25 learning models, learning systems, parameters, and/or variables are defined that describe various network and/or system properties. Such components can be adapted to determine statistical or pattern information regarding network and system activities. This information can include the number, quantity or average of network requests to and from one or more assets or network devices, the type of requests (*e.g.*, read/write, role/identity

of person/system requesting access, time of requests, location of requests), status or counter data (*e.g.*, network access counters, error codes), and/or substantially any data type or pattern that may be retrieved from a network, automation asset, or network device. At 1214, system learning is performed. This includes monitoring an automation network during a predetermined training period, wherein the learning components described above acquire information about network, system, user, and/or device activities during the training period. For example, a counter variable may learn the average number of network requests that are sent to an automation asset in a given time period (can also be other statistical measures than average). In another example, an intelligent component such as a Bayesian inference model, probability determination, or neural network learns patterns such as "During heaviest network loads, the PLC responds to 25% fewer requests, and during real time processing operations, 35% fewer requests for a maximum of 23 requests per minute processed during such periods, +/- 1 standard deviation."

After the training period at 1214, learned patterns are compared to current data patterns in view of predetermined threshold or range settings at 1218. For example, if the mean number of factory network packets transmitted is learned to be about 20,000 bytes per/second, +/- 5000 bytes, and a range is set up so that if network traffic goes above 26,000 bytes per second or below 10,000 bytes per second, then system security performance is considered acceptable as long as network traffic remains in the selected range. It is noted that thresholds/ranges can be set according to user desires, automated determinations, and/or according to the amount of risk and/or costs that are deemed acceptable (*e.g.*, for lesser amount of security risk, set thresholds closer to learned patterns).

At 1224, a determination is made as to whether or not deviations were detected from learned data patterns at 1218. If no deviations are detected, the process proceeds back to 1218 for further comparison processing. If deviations are detected at 1224, then one or more automated actions may be performed. Similar to the process described above, this can include automatically correcting security problems such as automatically adjusting security parameters, altering network traffic patterns, installing new security

components, removing suspicious components, firing alarms, and/or automatically notifying entities about detected problems and/or suspicions (e.g., sending an e-mail, alerting a pager, calling a number, generating a file, sounding an alarm, interrupting a web session, opening an instant messaging service, and so forth). After automated
5 processing at 1228, the process proceeds back to 1224 for further security processing, comparison, and detection.

What has been described above are preferred aspects of the present invention. It is, of course, not possible to describe every conceivable combination of components or methodologies for purposes of describing the present invention, but one of ordinary skill
10 in the art will recognize that many further combinations and permutations of the present invention are possible. Accordingly, the present invention is intended to embrace all such alterations, modifications and variations that fall within the spirit and scope of the appended claims.